DOCUMENT RESUME

ED 328 217 IR 014 812

AUTHOR Devi, Roshni
TITLE Machine Learning and Tutoring System. Centre for Information Technology in Education Report No. 61.
INSTITUTION Open Univ., Walton, Bletchley, Bucks (England). Inst. of Educational Technology.
PUB DATE Jan 89
NOTE 30p.; For related reports, see IR 014 806-808 and IR 014 811-813.
PUB TYPE Information Analyses (070) -- Reports - Research/Technical (143)

EDRS PRICE MF01/PC02 Plus Postage.
DESCRIPTORS *Algorithms; Computer Assisted Instruction; *Computer System Design; *Expert Systems; Foreign Countries; Instructional Systems; Man Machine Systems; *Models; *Programed Tutoring
IDENTIFIERS *Intelligent Tutoring Systems

ABSTRACT

This paper summarizes some of the current approaches to student modelling in Intelligent Tutoring Systems (ITS) and proposes ways in which learning algorithms can be applied in ITS. The role of machine learning in tutoring systems is described, including student modelling, teaching strategies, and collaborative learning. The PIXIE system and the Automated Cognitive Modeller system are examined as examples of systems using machine learning for student modelling. Dynamic student modelling is explored in the context of bounded user modelling technique, which not only identifies the student's knowledge at any one time, but also identifies the way in which the knowledge is acquired. Also discussed is the latest area of research in machine learning, namely, "explanation based learning" (EBL), in which explanations are constructed from domain knowledge and examples of the goal concept. The paper concludes with an outline for a learning algorithm for deriving concepts from subconcepts. (23 references) (DB)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

☐ This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

ED 328 217

Centre for Information Technology in Education

Machine Learning and Tutoring System

Roshni Devi

CITE Report No.61

Institute of Educational Technology
Open University
Walton Hall
Milton Keynes
Great Britain

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY

E.S. Henderson

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

MACHINE LEARNING AND TUTORING SYSTEMS

Roshni Devi
Institute of Educational Technology
The Open University
Walton Hall
Milton Keynes
MK6 7AA.

Jan. 1989

Abstract

This paper provides a summary of some of the problems with present approaches to student modelling. It outlines the potential roles of machine learning in Intelligent Tutoring Systems (ITS). Some systems that have applied machine learning for student modelling are reviewed. This is followed by a discussion on dynamic user modelling and a review of bounded-user modelling technique as a candidate for dynamic user modelling. The scope of machine learning research for this kind of modelling is highlighted. One such research topic of relevance in ITS is the derivation of concepts from subconcepts. My research explores the use of Explanation-based learning (EBL) for modelling this.

In the second half of the paper, EBL is reviewed, followed by an outline of the algorithm for the proposed research.

MACHINE LEARNING AND TUTORING SYSTEMS

1. INTRODUCTION

Research in ITS has focussed on creating student models. This has been done either by representing the student as a 'subset' of the expert ('subset' or 'overlay' modelling) or by using a library of bugs ('perturbation' modelling). Both approaches have problems in completely representing the student's knowledge. The former does not take into account the student's conception/representation of the domain, which may not necessarily be the same as the expert's. The latter requires extensive domain analysis in order to create a library of 'bugs'. In addition, neither of them take into account how these bugs are acquired. Hence, appropriate remediation cannot be expected.

In order to avoid costly domain analyses, machine learning techniques have been applied in the construction of bugs. This does reduce the amount of work needed to collect a library of bugs, but the basic problem of the conceptual details of the bugs still remains. In addition, there is also the problem of 'what to do with the student model' or 'how to make use of it for remediation or for deciding the next teaching action'.

Recent research in ITS is considering the use of several interactive styles. Examples are Self and Gilmore's (1988) idea of "colloborative learning" (reviewed in section 2.3) and Elsom-Cook and Spensley's (1988) idea of "multiple teaching strategies". A tutoring system with multiple teaching strategies must be capable of choosing one that is most appropriate at the time. Knowledge of the learning strategy used by the student (even at a shallow level, like 's/he probably applied analogy...') could be one of the contributing factors in making this decision.

The objective of my research is to construct a learning algorithm that can be used in a tutoring system for modelling the learning of concepts from subconcepts.

This paper outlines some of the potentials of machine learning in ITS. Two ITS that have used machine learning are reviewed. Section 4 proposes how learning algorithms can be applied in ITS to model the student's learning strategy. The bounded user modelling technique for dynamic modelling is also reviewed. In section 5, EBL is discussed. Finally, an outline of my

proposed research is presented.

2. THE ROLE OF MACHINE LEARNING IN TUTORING SYSTEMS

This section outlines the scope of machine learning for student modelling, for deciding teaching strategies and for supporting collaborative learning.

2.1 **Student modelling**

1. Elsom-Cook's (1987) *bounded user modelling technique* (reviewed in section 4) is one way in which machine learning has been used for student modelling. Since it is very difficult to construct a model of the student's exact knowledge state, Elsom-Cook proposed that we could construct the upper and lower bounds of the student's possible knowledge state such that the exact model lies somewhere in between. By improving the learning algorithm for constructing bounded user models, we can improve the resulting student models. For example, more psychologically plausible learning algorithms would give alternative bounds. Moreover, we could reduce the gap between the upper and lower bounds, hence reducing the number of possible models.

2. With machine learning research, we have the potential of being able to model the student's conception of the domain, i.e. we do not necessarily have to model the student in terms of the expert's representation. This could be achieved, for example, through research in machine learning on the program's ability to change its representation language.

3. Machine learning has been applied in ITS for automating the construction of student models. Examples of such systems are ACM (Langley et al., 1984) and PIXIE (Sleeman,1983), which are reviewed in section 3. The main aim of this application has been to avoid extensive libraries of bugs by providing the data and getting the machine to generate these bugs. The resulting student models are static models like those of LMS (Sleeman and Smith, 1981) and DEBUGGY (Brown and Burton, 1978).

2.2 **Teaching strategies**

1. Using machine learning, the tutor could learn about teaching strategies, by experimentation. For example, while teaching, the tutor could create chunks of knowledge like *If I teach concept X through method Y, then the*

learner learns it via method Z. In addition, the tutor could also learn which strategy is best suited for a particular individual.

2. Machine learning could also be applied for making decisions about what teaching action to take next. For example, if the student has learnt something in one way, the tutor could try teaching it in a different way, to allow the student to explore the different possibilities.
At the moment, all the knowledge about teaching strategies is explicitly represented in the system. With further research, we could have the system reason about strategies, so that it could derive its own teaching strategies, and learn when it is best to apply them.

Two systems that have applied learning to teaching strategies are Quadratic Tutor (O'Shea, 1979) and Proto-Teg (Dillenbourg, 1988). In the quadratic tutor, the tutorial strategy is represented as a set of production rules. The tutor improves by experimentation - by modifying the production rules. This is done by a database of assertions, where the assertions relate tutorial goals to the modifications that can be made to the tutorial strategy. To carry out an experiment, the tutor selects a tutorial goal, selects the assertions that relate to this goal and then executes the changes suggested by the assertions. After a student completes a lesson, the change in teaching performance is statistically evaluated. This step is repeated for seven students, and then the changes incorporated if the measure of overall improvement increases.

Proto-Teg is a tutoring system for geometry. This tutor has nine (slightly differing) teaching strategies. Its aim is to learn to select one of the given teaching strategies. It does this by learning the characteristics of the student model that are useful for selecting a teaching strategy. These characteristics are learnt by comparing the characteristics of the student model when the strategy was effective and those when it was not effective (This is done by applying an inductive learning technique). The operations carried out by the tutor can be summarised as follows:
1. Select and apply a teaching strategy
2. Update the (characteristics of the) student model.
3. Repeat steps 1 and 2 until the student masters the concept being taught.
4. Repeat steps 1 to 3 with other students (data from at least 6 students is needed before the learning algorithm can be applied).
5. Learn the conditions (characteristics of the student model) under which each teaching strategy is best applied, using the data obtained in step 2.

2.3 **Collaborative learning**

Machine learning also has a role in the use of tutoring systems for facilitating colloborative learning.
Self and Gilmore (1988) propose an approach to student modelling, which aims to model an 'artificial' student, who acts as a 'collaborative' partner of the 'real' student. The partner receives the same information as the student. The aim is for the partner to have a collaborative discussion with the student. A concept learning algorithm is applied to model the partner's learning strategies, and it is hoped that the student's learning strategies can be modelled as a "variation" of the program's (partner's) learning strategies. With further research on machine learning to create psychologically plausible learning algorithms than the existing ones, this approach has considerable potential for modelling learning strategies. As Self says:
"Effective collaboration would demand psychologically credible machine learning techniques and these are currently lacking." (Self, 1988. p.4)

The focus in this paper is on the first role above, i.e. machine learning for student modelling. (The proposal in section 4 is also applicable for the collaborative learning approach proposed by Self). The following section is a review of two systems which have used machine learning for constructing models of the student.

3. SYSTEMS USING MACHINE LEARNING FOR STUDENT MODELLING

3.1 **PIXIE**

PIXIE is an extension of a previous modelling system, called Leeds Modelling system (LMS). The system is in the domain of algebra. PIXIE has a library of rules containing the bugs (called malrules) and the correct rules. The student model is made up of rules from this library. To account for student errors which were not encountered before (i.e. there were no rules in the library corresponding to these errors), PIXIE was developed to infer new malrules.

This was done using the student's protocols, working backwards from the student's answer to the question. In the algebra domain, since there are several steps involved before the answer can be reached, and at each step, there are several alternatives that a

student can take, there is a huge space to search. While searching the paths from the answer to the problem, PIXIE first uses its library of rules. If the path can be constructed with correct rules (in the correct order), then the student's answer is correct. If they can be constructed using malrules and correct rules, then the student's errors can be explained in terms of the existing rules. The challenging problem is encountered when the search fails, i.e. the complete solution-problem path cannot be explained. In this case, PIXIE hypothesizes a new malrule which explains the last step in arriving at the question. The following example shows how this is done:

A) X = 3 (student's answer)
B) X = (3-n) + n (n=8)
C) X = 8 - 5
D) X + 5 = 8
E) X + (5-n) + (n) = 8
(Instantiate n using lhs of the question)
F) X + 2 + 3 = 8
G) 2X + 3 = 8 (question)

At each step in inferring the student's path, one of the following alternatives is applied:

i) One of the existing rules are applied, backwards. For example, step D in the above example is derived using the (correct) rule, *lhs +/- M = rhs ==> lhs = rhs -/+ M*, backwards (i.e. X + 5 = 8 <== X = 8 - 5).

ii) Focussing heuristics, selected from a set of about ten of them, are applied in order to get the equation to have the same form as the 'target' equation (i.e. the question). For example, at step B, the following heuristic rule is applied:

IF lhs(eqn) is not equal to lhs(target)
AND rhs(eqn) is not equal to rhs(target)
AND rhs=i
THEN replace (i, (i-n) + (n))

iii) If neither of the above can be applied, then a new rule is inferred. In the step from F to G in the above example, the system makes the inference that the student went from step G to step F. Hence, AX + B = C ==> X + (A + B) = C. Ignoring the symbols which are not changed, the system infers that the malrule that the student applied was AX ==> X + A.

There are two problems with this approach:
i) It would only work if the student's error is in the first step of his/her solution (in the above example, AX + B = C ==> X + (A + B) = C), which obviously is not general enough - it is equally likely for the student to make an error at any other step in the solution.
ii) A large number of heuristics (like the one at step B above) need to be provided. This leads to a huge search space.

On the one hand, PIXIE does not require a library of malrules to be provided before a student's malrules can be identified. On the other hand, it still needs some kind of 'domain analysis' - that of the set of heuristics.

3.2 **Automated Cognitive Modelling**

The Automated Cognitive Modeller (ACM, Langley et al, 1984) applies the machine learning technique of learning from examples to generate student models. The modeller is aimed to be a general tool for cognitive modelling, although the authors describe it mainly in the context of the domain of subtraction. The resulting model is a description of student behaviour like that produced by LMS.

ACM requires two basic inputs:
i) a problem space.
ii) some information about the behaviour of the student to be modelled.
The problem space consists of:
i) the representation of individual states (eg. for the domain of subtraction, the initial state is the problem, the final state is the solution to the problem).
ii) a set of operators for moving from one state to another (eg. add-ten, shift-column, decrement, etc.).
iii) a set of rules that state the conditions under which the operators may be applied (eg. the following is a condition-action rule for the operator add-ten

```
add-ten
If you are processing column1,
and number1 is in column1 and row1,
and row1 is above row2,
then add ten to number1).
```

The second input (student's problem-solving behaviour) consists of a set of problem-final answer pairs (note that PIXIE generates its malrules from a single problem-answer pair).

Given a student's answers to a set of problems and the initial set of condition-action rules, ACM needs to determine the sequences of operators that led to the student's answers and infer the conditions that will reproduce these sequences on the same problems. In other words, the student modelling task is reduced to the problem of determining whether a given operator was used by the student, and if so, determining the conditions under which it was used.

Using the initial set of general rules (provided as input iii) above), and the student's answer to a problem, ACM generates a search tree consisting of the observed (student's) answer along with many others. Figure 1 shows the search tree for determining a student's path for the problem 93 - 25 = 72 (This student has the 'smaller from larger' bug). Each step along the path leading to the student's answer is labelled '+' (a positive instance of the operator). Those steps leading one step off the solution are labelled '-' (negative instance of the operator). This process of searching the solution paths is repeated for the other problems. These solution paths provide a set of instances of operator applications. For example, examining the search tree in figure 2, there are two positive instances (5 - 3 and 9 - 2) and two negative instances (3 - 5 and 2 - 9) for the find-difference operator, one positive instance for the shift-column operator, two negative instances for add-ten operator and one negative instance for the shift-left operator.

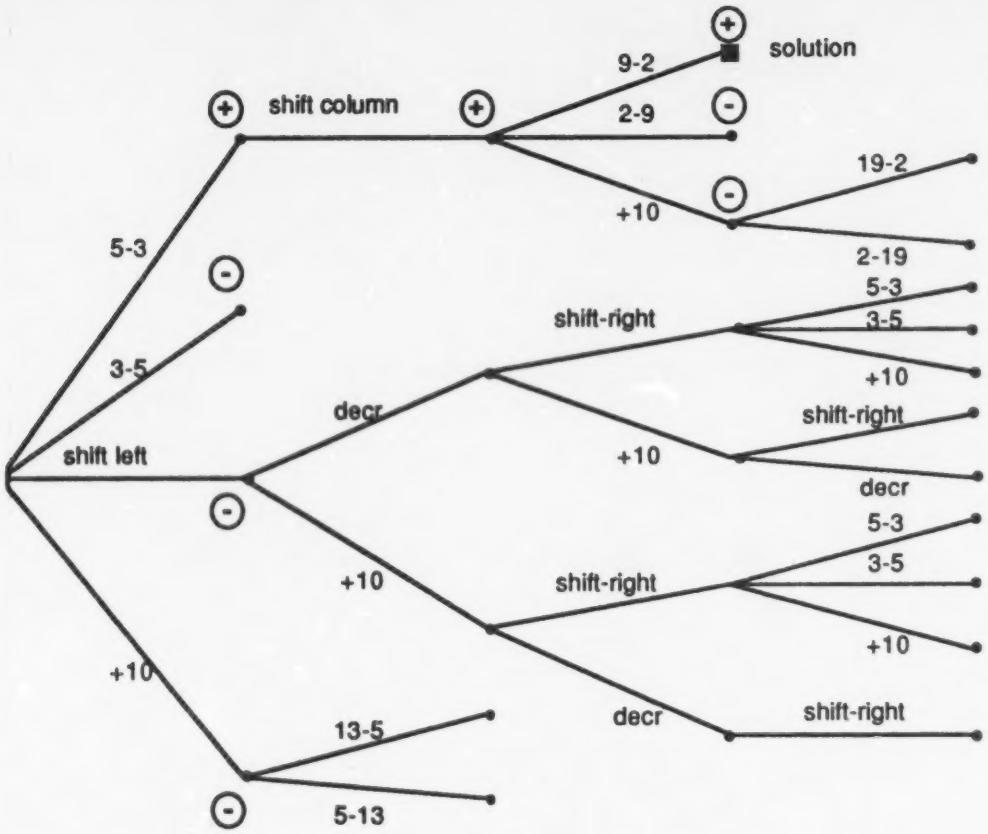


Figure 1 ACM's search tree for the problem 93 - 25 = 72 (Wenger, p. 214)

Having obtained a cumulative set of positive and negative instances for each operator from the search trees of the set of problems, the conditions under which the operator is applied are determined using a discrimination learning method (For details of discrimination learning method, see Devi, 1988).

For determining the conditions, a set of tests have to be provided (eg. greater N1 N2, above R1 R2). The tests are used for discriminating positive and negative instances of an operator. Those tests, that satisfy only positive instances of an operator, make up the conditions of the operator applications (such conditions-operator rules make up the production-system model of the student's behaviour).

Though ACM's approach to modelling does get rid of 'extensive domain analyses', it still needs knowledge of the domain. Examples of such knowledge are:

i) the problem space - the operators and the overly general production rules.
ii) lists of tests (eg. greater N1 N2).

ACM's modelling capability depends on the initial domain knowledge provided, which in turn depends on some knowledge of malrules in the domain. This is because the operators have to account for all possible actions that students may take. Another limitation of ACM's modelling technique is that it would work only for structured domains, or for domains in which the intermediate steps that a student can take are known. In addition, due to the huge search space, the approach would be practical only for those domains which have a small number of ways of arriving at the solution (i.e. a small number of operators).

Both the systems above are diagnostic systems, not implemented for tutoring. Neither of them say anything about how the student models will be used for tutoring. Presumably, if one wishes to implement a tutoring system using these approaches to modelling, one has to go through two phases:
I) the diagnostic phase
ii) the tutoring phase plus the diagnostic - this would be implemented using the student models obtained in phase i.
The second phase still includes the diagnostic phase. This is to discover any new malrules that have not been encountered in phase i. (The appropriate remediation corresponding to this new malrule will then have to be programmed in the tutoring system).

In addition, neither of the modelling techniques attempt to model the learning process of the student. Hence, they cannot explain how the malrules were first acquired.

Both the approaches are general enough to be applied to domains which are structured and those which have a small number of possible ways of problem solving.

4. TOWARDS DYNAMIC STUDENT MODELLING

An approach to a more complete model of a student would be the *dynamic* modelling of the student's learning which not only identifies his/her

knowledge at any one time, but also identifies the *way* in which that knowledge is acquired. Two ways in which this could be achieved, are as follows:

i) While tutoring, the tutor could apply a learning algorithm to the concept being taught in order to imitate the way the student might be learning it (This is similar to Self's idea of learning by a 'collaborative partner', and Elsom-Cook's approach to bounded user modelling). Such an application of learning would allow the system to make predictions about what concepts the student might have acquired and about *how* they might have been acquired. Since no single learning mechanism that is general enough for this purpose exists at the moment, the next approach might be preferred.

ii)The system could have several learning algorithms. It could apply each one to the concept being learnt and compare the results of its learning with that of the student's. Some heuristics could be implemented for choosing between the learning strategies to apply. One possibilty of this could be analyzing the knowledge that is available for learning; for example, if analogy can be drawn to some previous knowledge, then try applying analogy. The information available for the system's learning is the same as that available to the student. In most domains, there are several possible paths that a student could choose while learning or doing problem solving. The system could interact with the student to gather information in order to make predictions about the path it should take (However, this has to be done in such a way that the student does not get bored with repeated testing). If a learning algorithm's result matches that of the student's result, then the algorithm is a good approximation of the student's learning mechanism(s). This modelling technique would also be able to take into account individual differences in the choice of strategies applied. (For the same domain, different individuals might choose different learning strategies).

Such dynamic modelling would enable the tutoring system to make more informed predictions about teaching strategies it should use. For example, the system could have knowledge about teaching strategies associated with learning strategies. Once the system has the ability to predict the student's learning strategy, it can choose the teaching strategy accordingly.

Bounded user modelling

The bounded user modelling technique is an example of dynamic

modelling. It is proposed that the technique can lead towards making predictions of the student's reasoning/learning process. This is achieved by applying a model of the student's learning process on the concept that s/he is trying to learn. The technique is based on Mitchell's version space algorithm (1981, see Devi, 1988 for a review). Elsom-Cook (1987) argues that in a tutoring system, at each event, a number of things may be learnt, hence the learning algorithm must be capable of learning a number of concepts at once. This can be achieved by having a learning algorithm that maintains a set of version spaces. This approach has several advantages:

i) Ideally, it will be able to model an individual's learning of more than one concept at a time (if required).

ii) It can also account for individual differences in the concept(s) learnt at a particular event. For example, let us assume that there are two possible concepts to be learnt (i.e. two version spaces). Student A might learn concept 1; Student B might learn concept 2; Student C might learn both concepts 1 and 2, etc.

iii) Within the process of learning a concept, the version space approach is suitable for representing undergeneralization and overgeneralization errors (These are common causes of errors). For example, while learning commutativity of addition, children often undergeneralize the principle to small numbers only. Hence, 10 + 7 would obviously be the same as 7 + 10, but 70 + 100 and 100 + 70 would not be the same to a number of children. For the same principle, a common overgeneralization error would be generalizing commutativity to subtraction. Hence, 5 - 6 is the same as 6 - 5.

iv) The bounded user modelling technique has the potential of applying the learning strategy that the student applied, at each example-stage in the process of learning the concept. (This requires psychologically plausible learning algorithms). It seems plausible, too, that we apply single learning strategies at a single snapshot of the learning process, and different strategies may be applied at different snaphots in the process. For example, the student might apply analogy between the example and previous knowledge (where previous knowledge could be the first example, or the generalization of the concept obtained after the first example - these could be recognized by using machine learning analogy algorithms). S/he could have learnt a subconcept after the first example, then learnt the concept after the second example using the subconcept as prerequisite knowledge (The concept learning algorithm proposed in section 6 would be ideal for recognizing this learning strategy). Once the system has these learning

algorithms at its disposal, it could apply the appropriate one at each stage of the concept learning task to make predictions about the student's learning.

The bounded user modelling technique has the potentials of modelling what has been learnt as well as how it has been learnt. Research in machine learning obviously has much to contribute to improve on this technique. For example,

i) In constructing more psychologically plausible learning algorithms than those that exist at present.

ii) In the program's ability to change its representation language. This would allow the system to represent those descriptions of a concept which could not be represented in the system's previous representation language.

5. MACHINE LEARNING

One of the goals of research in machine learning and in ITS is to model human learning mechanisms. This has obvious implications for education (standard classroom teaching) and for the future design of ITS. J. Self remarks that

"The study of human learning processes (whether by computer or not) has practical significance in that any progress may lead to more effective educational techniques." (Self, 1987, p.1).

"*In order to present material at a level of difficulty and detail suited to the state of knowledge of the student, the system must know and follow the student's changing knowledge*" (Michalski, 1986, p.7).
This would mean that the system has to have a model of the student's learning process, and to achieve this, machine learning seems to be the ultimate solution.

Several learning strategies have been identified in machine learning. Our interest is in the strategy of *learning from examples* (also called concept acquisition). In learning from examples, the aim is to learn a general description of the target concept from a set of positive and negative examples of the concept (For more details on this learning strategy, the reader is referred to the previous document). In this paper, the focus is on a sub-strategy of learning from examples, called *Explanation based learning* (EBL).

Explanation-based learning

The latest area of research in machine learning is *Explanation based learning* (EBL). In EBL, explanations are constructed from domain knowledge and a positive training instance of the goal concept (Positive instance is an example of the concept, i.e. it is a member of the class of objects that make up the concept. In contrast, a negative example is a counter example). An explanation is a proof of the example being a positive example of the concept. The features of the example used in the explanation are then used to obtain a generalization of the concept. This is done by the process of goal regression (Waldinger, 1977).

In EBL no new knowledge is acquired. The domain knowledge contains the concept under study, and the goal of EBL is to redefine the concept in operational (useable) terms. In other words, EBL restructures or reformalizes existing knowledge.

"They are explanation-based in the sense that they construct explanations of why an object satisfies a functional definition, and use this explanation to construct a structural description." (Carbonell & Langley, 1986, p.25).

Mitchell, Keller and Kedar-Cabelli (1986) propose a generalization/ unification of previous EBL methods. This unified approach, called *Explanation Based Generalization* (EBG), requires four explicit inputs:

1) A training example.
2) A domain theory.
3) A definition of the concept under study.
4) A description of the form in which the learnt concept must be expressed. (This is referred to as the *operationality criterion*).

The domain theory contains information on how to transform the third input into a definition of the concept that satisfies the fourth input.

There are two steps in the EBG process:

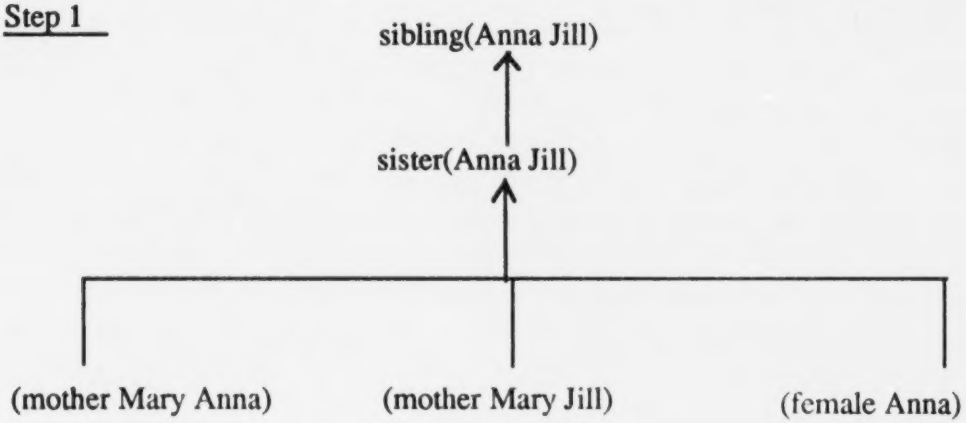
1) *Explain* - construct an explanation using the domain knowledge to prove that the example is an example of the goal concept.
2) *Generalize* - determine a set of conditions under which the explanation of 1) holds, stated in terms of the operationality criterion. In other words, find the most general version of the explanation consistent with the domain theory.

To see more clearly how this works, let us consider a very simple example.

Goal concept: *The concept of 'sibling'*
Training example: *(female Anna)*
(mother Mary Anna)
(mother Mary Jill)
Domain theory: *(brother x y) V (sister x y) --> (sibling x y)*
(mother x y) & (mother x z) & (female y) --> (sister y z)
(mother x y) & (mother x z) & (male y) --> (brother y z)
Operationality criterion: *The concept definition must be expressed in terms of the predicates used to describe the example.*

Note that it is essential to have at least one rule in the domain theory with 'sibling' on the right-hand side. It is also essential for the domain theory to have some rules whose left-hand sides are predicates in the training example.

Step 1



Step 2 The operational definition of sibling is
(mother x y) & (mother x z) & (female y) --> (sibling y z)

Step 2 is achieved by goal regression through the explanation tree, generalization of constants to variables and unification of variables.

The example above illustrates some of the limitations of EBG:

1) The generalization obtained from one example only can be biased or specific to that particular training example. This weakness has been pointed out by Mooney and DeJong (1985) as well.
"It often does not generalize the new concept far enough from the particular training

example. The result is undergeneralizations that reflect many unimportant details of the example problem." (DeJong and Mooney, p.146).
This limitation will be overcome in the proposed model (next section) by considering other examples.

2) The algorithm is not completely successful with disjunctive concepts (as illustrated by the disjunctive concept above). The goal concept is a disjunction. The explanation is constructed (the example is justified) when one of the disjunctions is satisfied. As a result, the operational concept description is made of this one disjunction only. In the example above, the disjunctive concept is *brother(x y) V sister(x y)*. The training instance satisfies *sister*, and hence the resulting generalization does not say anything about *brother*. However, this method is able to represent disjunctive concepts (in the functional definition of the goal concept) which is a limitation of most other learning methods.

3) It is likely that the generalization will be only a subset of the goal concept. In the above example, this is the case - the concept defines only a female as being a sibling of someone else. Hence, *(sibling Tom Anna)* would not be true according to the concept definition obtained. This is partly due to the first two limitations.

4) It requires significant domain knowledge.

5) The resulting generalization is only one of the several possible generalizations.

Despite the limitations, the idea of using domain knowledge to construct explanations provides possibilities for future. In addition, EBL, though it requires some improvements, is one of the most psychologically plausible strategies of learning. The use of domain knowledge is quite similar to activating the relevant schema in memory when learning something in a new domain. In addition, it seems plausible that human beings need to restructure their knowledge depending on what is being learnt. (This is equivalent to accommodation of knowledge). It also seems plausible that we learn concepts from a single example, if we have the prerequisite subconcepts. Ahn, Mooney, Brewer and DeJong (1987) provide evidence of psychological validity of EBL.

Some other advantages of this approach are:
- a justification is provided for the concept description that is generated.

- it can learn from a single example. Multiple examples (and negative examples) are not required.
- it is capable of handling noisy data, since these will be identified in the explanation process. Provided the domain knowledge is complete, if an explanation cannot be constructed, it would imply that the data (instance) is noisy.

6. RESEARCH PROPOSAL - USING EBL TO LEARN CONCEPTS FROM SUBCONCEPTS

The objective of the research is to implement a learning algorithm for derivation of concepts from subconcepts. The algorithm is based on EBL. After applying EBL to the first example, one or more generalizations of the concept to be learnt may be obtained. The generalization(s) (and the tutoring system's knowledge of the goal concept) will be used to select the next example, in order to refine any previous generalization(s) and/or to reduce the set of possible generalizations.

This strategy of learning makes use of domain knowledge. Previous EBL programs (eg. Mitchell, Keller & Kedar-Cabelli 1986) use domain knowledge to prove that a positive instance is an instance of the concept. The proposed model uses domain knowledge slightly differently - domain knowledge represents the previous/existing knowledge of the domain, which acts as the foundation or the prerequisite for the concept to be learnt. In Mitchell et al's program, the domain knowledge contains the concept, whereas in the proposed model, since the concept is to be learnt, it is not contained in the set of existing knowledge. Instead, we assume that the domain knowledge contains a prerequisite subconcept. This approach would allow the learning system to learn concepts that are more complex than it already knows.

"*The meaning of human concepts is dynamic; it changes with time and adapts to new contexts and requirements. Novel concepts are continuously being created and developed,... We need a learning system capable of acquiring new concepts and concept structures by generalization from examples or by analogy to prior knowledge. Such a system should be able to modify, specialize, or generalize old concepts in a flexible fashion.*" (Michalski, 1986, p.7).

Outline of the algorithm

(Details of the algorithm can be found in Devi, forthcoming).

The following are the steps in the proposed EBL algorithm:

1. Get the first (positive) example (as input).
2. Retrieve relevant domain knowledge. (For this step, I need to consider the mechanisms of retrieval from memory.)
3. Combine the example and the domain knowledge to get a more abstracted description of the example.
4. Repeat steps 2 and 3 to generate all possible abstractions.
5. Collect all the abstractions.

6a). If there is a single one-element spanning set, then the concept has been identified.

6b). If there is more than one one-element spanning set, then the concept is ambiguous. There are several alternatives:
   i) Compare the sets, eg. by applying a similarity generalization.
   ii) Combine/generalize the sets (eg. by adding disjunction(s)).
   iii) Go on to the next stage - get the next example.

6c). If there is no one-element spanning set, then the alternatives are:
   i) Carry all spanning sets to the next stage - get the next example.
   ii) Use some heuristics for choosing one favourite set.
   iii) Use some heuristics for eliminating unlikely sets.
   iv) Do some generalization on the sets.

The following is an example of applying the above algorithm in the process of learning the concept *sibling*:

Domain knowledge:

(male Tom) (female Anna)
(father Peter Tom) (father Peter Anna)
(mother Mary Tom) (mother Mary Anna)
(male Peter) (female Mary)
(mother Sally Sue) (mother Sally Jane)
(female Sue) (female Jane)
(mother Mary Jill) (female Jill)
(mother =x =y) (mother =x =z) (female =y) --> (sister =y =z)
(mother =x =y) (mother =x =z) (male =y) --> (brother =y =z)
...

Example: (Anna Jill)

At step 5, two abstractions are obtained:

Abstraction1: (sister Anna Jill)
Explanation1: (female Anna) (mother Mary Anna) (mother Mary Jill).

Abstraction2: (sister Jill Anna)
Explanation2: (female Jill) (mother Mary Anna) (mother Mary Jill).

These results correspond to step 6b) above - the concept to be learnt is ambiguous, i.e. it could be *sister(x y)* or it could be *sister(y x)*. (At this stage, the learning algorithm could learn that sister relation is commutative). Applying each of the alternatives listed in 6b:

i) comparing the explanations, the following are common (analogous to applying the *dropping condition* generalization rule):
a. *(mother z x) (mother z y)*
b. *(female x)*

ii) using the *adding disjunction* generalization rule, the following generalization can be obtained:
(sister x y) v (sister y x)

iii) worry about refining the descriptions after seeing the next example.

The results of the EBL (i.e. generalization(s) and explanations) stage are carried over to the 'next-example stage' of learning. At this stage, the tutoring system can look at the student's concept description and the possible explanations for the concepts, and then decide the next example. For the above example domain, the system can reason, for example, as follows:
the concept that I am trying to teach is sibling (which is sister or brother); the example I have shown is one where both persons are females (from i b above); let me select an example with a male.
Hence, the next example could be *(Tom Anna)*.

Why is this approach to combining EBL and learning from examples interesting?

Firstly, in a tutoring system, information gathered from the student-tutor interaction can be used as the domain knowledge required for making predictions or explanations (for applying EBL). In addition, EBL has the advantage that the generalizations obtained can have explanations or annotations attached to them, which provides the reasons for making a particular generalization. Hence, in a tutoring system, the tutor can look back and see why it made the prediction

(s) that it did about the student as well as why the student may have learnt what s/he did. To refine the prediction(s) made from the first example,

the next example can be used. While selecting the next example, on-line, based on the educational objectives (like increase/decrease the difficulty level), the tutor can have an additional objective - that of reducing the set of generalizations/ predictions that it made after the first example, and reinforcing/refining the reduced set.

Secondly, the fact that EBL is applied first resembles human learning of a new concept/domain. In the early learning of a new concept, the schema for previous knowledge of the domain is activated. It is also consistent with classroom learning where the teacher often helps to activate the right schema (for example by revision of previous lesson/ prerequisite knowledge). Using this strategy, the acquisition of new knowledge can be guided by what the learner already knows. In addition, new (or more complex) concepts can be learnt after making use of the already learnt concepts.

Thirdly, the advantages of EBL and that of learning from examples is added to the combined strategy. For example, the likelihood of the concept description being overspecific and the problem of finding the most general description in EBL are overcome by the learning from examples strategy.

Fourthly, there are many possible generalizations of the concept, and hence learning from examples applied alone would need to search a huge space of generalizations. EBL constrains this search.

Fifthly, this approach is more psychologically plausible than the other approaches to combining EBL with learning from examples that are currently being undertaken by machine learning researchers. The following is a brief description of the other approaches:

1) Applying learning from examples to obtain a set of possible generalizations. This set is then pruned and refined using EBL, to obtain an explained/justified generalization (explained in terms of domain theory). An example of a system utilizing this approach is UNIMEM (Lebowitz, 1986).

2) Applying EBL to each example (which is computationally expensive) and learning from examples to the resulting generalized examples. Such an approach is employed in the WYL program (Flann and Dietterich, 1986).

7. FUTURE WORK

1. The learning algorithm will be implemented and tested for its performance on both artificial and real domains.

2. Empirical studies will be carried out to compare the behaviour of the program with that of human learning of the same concept. For example, what generalizations do humans come up with after being given the required subconcept and an example of the concept? What are the limitations of the algorithm, in terms of not being able to represent all the details of what goes on in a human's mind.

3. The performance of the program can be compared with that of other learning programs like SOAR.

4. An analysis of how teaching and learning strategies can be incorporated in ITS. More specifically, an analysis of the implemented learning algorithm's use in ITS.

5. Theoretical analysis of the capabilities and power of the algorithm.

GLOSSARY

EBL - inferring a general description of the target concept from a single positive example and domain knowledge.

Generalization of an example - concept definition which describes a set containing that example.

psychological plausibility - 'human-like' (and this is at a surface level only as opposed to psychological validity which refers to a much deeper level comparisons, like memory limitations).

Learning from examples -inferring a general description of the target concept from a set of positive and negative examples.

Negative example - (usually in the context of learning from examples). Counter-example of the target concept.

Positive example - (usually in the context of learning from examples). An example or instance that is a member of the class of objects that make up the target concept.

Structural description - A symbolic representation for objects and concepts based on descriptions of their parts and the relationships among them.

Acknowledgements

I wish to acknowledge my supervisor, Mark Elsom-Cook for his continuous guidance, and for his contributions to the ideas in this report.
I am also grateful to Richard Joiner, Ben du Boulay and Rick Evertsz for their constructive comments on the first draft.

REFERENCES

Ahn, Mooney, Brewer and DeJong (1987) . Schema Acquisition from one example: Psychological evidence for explanation-based learning.

Brown, J. S. & Burton, R. R. (1978). Diagnostic models for procedural bugs in basic mathematical skills. *Cognitive Science*, vol. 2, p.155-191.

Carbonell, J. G. & Langley, P. (eds., 1986). Proceedings of the Learning and knowledge acquisition workshop.

Devi, R. (1988). A comparison of approaches to learning from examples. Cite report.

Devi, R. (1989, forthcoming). Design specifications of a concept learning algorithm. Cite report.

Dillenbourg, P. (1988). A pragmatic approach to student modelling: Principles and architecture of Proto-Teg. Paper presented at ITS conference in LeMans.

Elsom-Cook, M. (1987). Guided discovery tutoring and bounded user modelling in Intelligent Computer Aided Instruction. CITE report no. 13.

Elsom-Cook, M. & Spensley, F. (1988). Dominie: Teaching and assessment strategies. CITE report no. 37.

Flann, N. & Dietterich, T. (1986). Selecting appropriate representations for learning from examples. In *Proceedings of the national conference on artificial intelligence*. American Association for Artificial Intelligence.

Langley, P., Ohlsson, S. & Sage, S. (1984). Machine learning approach to student modelling. Technical report CMU-RI-TR-84-7. Robotics Institute, Carnegie-Mellon University, Pittsburgh, Pennsylvania.

Lebowitz, M. (1986). Concept learning in a rich input domain: generalization based memory. In R. S. Michalski, J.B. Carbonell & T.M. Mitchell (Eds.) *Machine Learning: An artificial intelligence approach*, vol. 2. Los Alto, CA: Morgan Kaufmann.

Michalski, R. S. (1986). Understanding the nature of learning: Issues and research directions. In R. S. Michalski, J.B. Carbonell & T.M. Mitchell (Eds.) *Machine Learning: An artificial intelligence approach*, vol. 2. Los Alto, CA: Morgan Kaufmann.

Mitchell, T. M. (1982). Generalization as search. *Artificial Intelligence*, 18 (2), 203-226.

Mitchell, T. M., Keller, R. & Kedar-Cabelli, S. (1986). Explanation-based generalization: A unifying view. *Machine Learning*, 1, 47 - 80.

Mooney, R. & DeJong, G. (1985). Explanation-based learning: an alternative view. *Machine Learning* 1, 145-176.

O'Shea, T. (1979). A self-improving quadratic tutor. In Sleeman, D., Brown, J. S. (eds.) Intelligent Tutoring Systems, 1982. London, Academic Press, 309-336.

Self, J. (1987) Human and Machine Learning. CeRCLe Technical Report No. 42, University of Lancaster.

Self, J. (1988). The use of belief systems for student modelling. CeRCLe Technical Report, University of Lancaster.

Self, J. & Gilmore, D. J. (1988). The application of machine learning to intelligent tutoring systems, in Self, J. (ed.), Artificial Intelligence and Human Learning, London: Chapman and Hall.

Sleeman, D. & Smith, M. J. (1981). Modelling students' problem solving. *Artificial Intelligence*, vol. 16, p. 171-187.

Sleeman, D. (1983). Inferring student models for intelligent computer-aided instruction. In R. S. Michalski, J.B. Carbonell & T.M. Mitchell (Eds.) *Machine Learning: An artificial intelligence approach*, vol. 1. Los Alto, CA: Morgan Kaufmann.

Waldinger, R. J. (1977). Achieving several goals simultaneously. In machine intelligence 18, p.94-136.

Wenger, E. (1987). Artificial intelligence and tutoring systems, Computational and cognitive approaches to the communication of knowledge. Morgan Kaufmann Publishers, Inc., CA.

CENTRE FOR INFORMATION TECHNLOGY IN EDUCATION

List of CITE Reports

These reports may be obtained from:

Hansa Solanki, Institute of Educational Technology, The Open University, Walton Hall, MILTON KEYNES, MK7 6AA, England.

Report No.	Title and Author
1	A.T. Vincent, (1985) Computing and the Blind.
2	A. Jones, G. Kirkup, J. Morrison (1985) A Trial of Home Based Computer Terminals.
3	Gill Kirkup, (1985) The Present and Potential Use of Ceefax in the Open University.
4	Mark Elsom-Cook, (1986) Artificial Intelligence and Computer Assisted Instruction.
5	Mark Elsom-Cook, (1986) A Pascal program checker.
6	Simon Holland, (1986) How computers are used in the teaching of music and speculations about how Artificial Intelligence could be applied to radically improve the learning of composition skills.
7	Simon Holland, (1986) Design consideration for a human-computer interface using 12-tone three-dimensional harmony space to aid novices to learn aspects of harmony and composition.
8	Alison Petrie-Brown, (1987) The Influence of Context and Coherence as a Foundation for Dialogue Research.
9	Eileen Scanlon, Randall B. Smith (1987) A Rational Reconstruction of a Bubble Chamber Simulation Using The Alternate Reality Kit.
10	Mark Elsom-Cook, (1987) Intelligent Computer-Aided Instruction research at the Open University.
11	Mark Elsom-Cook, (1987) Towards a framework for human-computer discourse.
12	Mark Elsom-Cook, (1987) MATILDA AND IMPART: Lisp tools.
13	Mark Elsom-Cook, (1987) Guided discovery tutoring and bounded user modelling in Intelligent Computer Aided Instruction ed. J. Self, Chapman-Hall 1987.
14	A.M. Petrie-Brown and M.T. Elsom-Cook, (1987) An Examination of an AI model of indirect speech acts.
15	A. Edwards, (1987) Integrating Synthetic Speech With Other Auditory Cues In Graphical Computer Programs For Blind Users.
16	S. Holland, (June 1987) A knowledge-based tutor for music composition.
17	S. Holland, (June 1987) New Cognitive Theories of Harmony Applied To Direct Manipulation Tools for Novices.

18 M. Baker. (July 1987) Intelligent Computer-Aided Instruction and Musical Performance Skills.

19 M. Baker, (August 1987) Proposed Research Directions for Intelligent Computer-aided Instruction in Musical Performance Skills.

20 A D N Edwards, (August 1987), Adapting interfaces for visually disabled users.

21 M Elsom-Cook, (September 1987), Acquisition of computing skills.

22 M. Baker, (September 1987), Computational Analysis of musical grouping structures.

23 M Baker, (June 1987), Automated Analysis of Musical Grouping Structures as a Basis for a Guided Discovery Environment for Interpretation of Music.

24 D Laurillard, (October 1987), The different forms of learning in psychology and education.

25 A.D.N. Edwards, (November 1987) Modelling blind users" interactions with auditory computer interface.

26 P. Fung, (November 1987) Novice Prolog Programmers.

27 P. Fung, B. DuBoulay & M. Elsom-Cook, (November 1987), An initial taxonomy of novices' misconceptions of the Prolog interpreter.

28 G. Kirkup, (November 1987), Considering the effect on women students of an increased use of microcomputers in distance education.

29 Sara Hennessy, Rick Evertsz, Dave Ellis, Phil Black, Tim O'Shea, Ann Floyd, Design Specification for 'Shopping on Mars' a computer-based Educational Activity.

30 A D N Edwards, The Use of home computers by disabled students at the Open University. Part 1: Previous use of computers in courses

31 Michael E. Fox (March 1988) A Report on Studies of Motivation, Teaching and Small Group Interaction with special reference to Computers and to the Teaching and Learning of Arithmetic.

32 Michael E. Fox (March 1988) Theory and Design for a Visual Calculator for Arithmetic.

33 M. Baker (February 1988) A Cognitive Model for Perception of Musical Grouping Structures

34 M. Baker (February 1988) An Artifical Intelligence Approach to Musical Grouping Analysis

35 P. Fung (December 1987) Novices' predictions of Prolog's control flow: A report on an empirical study

36 M. Elsom-Cook and F. Spensley (April 1988) Knowledge representation in a tutoring system for procedural skills.

37 F. Spensley & M. Elsom-Cook (April 1988) Dominie: Teaching and Assessment Strategies.

38 M. Baker (May 1988) Tutoring with Incomplete and Uncertain Knowledge

39 R. Moyse (May 1988) Multiple Viewpoints for Intelligent Tutoring Systems.

40 P. Whalley (May 1988) Hyper Technic - a graphic object-oriented control environment.

41 P. Whalley (June 1988) Cued recall as a Measure of argument integration.

42 L. Alpay (June 1988) A Survey and Examination of Intelligent Tutoring Systems in Medicine

43 M. Elsom-Cook (July 1988) Introduction to the ECAL system.

44 F. Spensley (July 1988) Dominie: Trainer Interface.

45 M. Baker (July 1988) An Architecture of an Intelligent Tutoring System for Musical Structure and Interpretation.

46 E.Scanlon and R.Sibbitt (April 1988) Proceedings of CAL Conference 88.

47 L. Alpay (September 1988) Medical Problem-Solving and Intelligent Tutoring Systems: Proposed Research Directions.

48 S.A Cerri, M.T. Elsom-Cook, M. Leoncini (September 1988) TRILL: The Rather Intelligent Little Lisper.

49 R. Evertsz, S. Hennessy & R. Devi (September 1988) GADL: a Graphical Interface for Mental Arithmetic Algorithms.

50 P. Fung, (October 1988) A formalisation of novices' errors in Prolog programs.

51 M. Baker, (October 1988) Arguing with the tutor: a model for tutorial dialogue in uncertain knowledge domains.

52 (October 1988) Information Technology in Education: Conceptual change in Science.

53 T. O'Shea, C. O'Malley & E. Scanlon (October 1988) Magnets, Martians and Microworlds: Learning with and Learning by OOPS.

54 (November 1988) Directory of Research 1988.

55 P. Fung (November 1988) Automated Diagnosis of Prolog control flow errors: a first evaluation.

56 R. Mason (December 1988) The use of Computer-Mediated Communication for Distance Education at the Open University, 1988.

57 A.D.N. Edwards & P. Grove (December 1988) A User's Guide to the Voice Screen Reader.

58 R. Sibbitt (December 1988) How Children Spell Nonsense Words.

59 Directory of Research 1988-1989.

60 R. Joiner (February 1989) Mechanisms of Cognitive Change in Peer Interaction: A Critical Review.

61 R. Devi (February 1989) Machine Learning and Tutoring System.